Improved Non-linear Image Enhancement for Video Coding

Lung-Jen Wang*1

Department of Computer Science and Information Engineering, National Pingtung Institute of Commerce, Taiwan
51 Min Sheng E. Road, Pingtung 900, Taiwan, R.O.C.

*1ljwang@npic.edu.tw

Abstract

It is well known that the B-spline filter can yield a very accurate algorithm for smoothing. In this paper, it is shown that a cubic B-spline filter can be used to improve the non-linear image enhancement method. In the non-linear image enhancement, a higher-frequency component can be predicted to solve the blurred problem of an enlarged image. This paper also presents a new three-dimensional (3-D) down-scaling scheme to subsample video data for video coding. Furthermore, a novel non-linear image-enhancement compensation algorithm with cubic B-spline filter is presented to improve the prediction of higher-frequency component and accordingly the efficiency of the 3-D down-scaling video coding. Finally, a computer simulation shows that the proposed method yields a better quality of the decoded image than other non-linear enhancement methods.

Keywords

Cubic B-spline Filter; Non-linear Image Enhancement; Video Coding

Introduction

With the growing interest of digital image processing, the applications in this domain such as digital high definition television (HDTV), Internet protocol television (IPTV), and video-phone, an integral part of our life are related to a scale image enlargement technique. Typically, however, the image enlargement causes a blurred image because there is no power in the high-frequency component of enlarged image [14]. To improve the quality of such a blurred image, the image enhancement is an indispensable post-processing method.

Image Enhancement is a very important topic in the researches. The principle of image enhancement is to process an image so that the result is more suitable than the original image in many applications. A typical image enhancement is achieved througth the high-pass filter followed by the post-processing in order to make the image suitable. In other words, this method uses a typical principle behind un-sharp masking and high-boost filtering [14].

Non-linear image enhancement [1, 2, 13] is similar to the typical image enhancement except that the high-pass filter is replaced by non-linear operations. This enhancement method uses the Gaussian-pyramid [1] or filter subtract and decimate (FSD)-pyramid [2] representation of an image to extract the high-frequency component from input (blurred) image. That is, a high-frequency component L_{-1} can be obtained from a blurred image by a nonlinear filter as shown in Fig.1. A new output image is generated next as the sum of the given input image and high-frequency component L_{-1}. The major non-linear step involves clipping and scaling the extracted components.

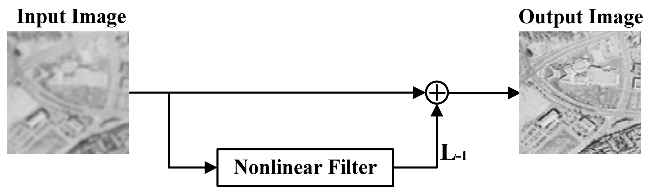


FIG. 1 THE NON-LINEAR IMAGE ENHANCEMENT

The cubic B-spline filters have been extensively used in image processing [3, 6, 7, 8, 10]. In [11] part, which is related to an improved non-linear image enhancement method for video coding, is proposed. However, the detailed derivation of both non-linear image enhancement with the cubic B-spline filter and the 3-D down-scaling video coding has not been presented. In addition, this paper gives more detailed descriptions on the non-linear image enhancement compensation with the cubic B-spline filter and the 3-D down-scaling video coding. Furthermore, the novel non-linear image enhancement compensation is used along with video coding to improve the quality of decoded image. Finally, some experimental results show that the proposed method obtains a better

subjective quality and objective PSNR performance than other non-linear image enhancement methods.

The rest of this paper is organized as follows. Section 2 describes the cubic B-spline filter. The non-linear image enhancement is proposed in Section 3. Section 4 shows the 3-D down-scaling scheme for video coding. In Section 5, the video coding using non-linear image enhancement compensation is described. The computer simulation is illustrated in Section 6. The last section shows the conclusions of this paper.

Cubic B-Spline Filter

Splines are piecewise polynomials with pieces that are smoothly connected together. The B-splines (where the B may stand for basis or basic) are the basic building blocks for splines [8]. In addition, it is shown in [3, 6, 7, 10] that the B-spline is a very good low-pass filter for image representation.

Let $\xi : \xi_0 < \xi_1 < \cdots < \xi_n < \xi_{n+1}$ be a partition of the interval [ξ_0 , ξ_{n+1}] on a real axis. The spline basis function of degree n on ξ in [3] is the following piecewise polynomial:

$$B_n(\xi;\xi_0,\xi_1,\xi_2,\cdots,\xi_{n+1}) = (n+1)\sum_{k=0}^{n+1}\frac{(\xi-\xi_k)^n U(\xi-\xi_k)}{\omega(\xi_k)} \quad \text{, for } n=0,1,2,\cdots \tag{1}$$

where

$$\omega(\xi_k) = \prod_{\substack{j=0 \\ j\neq k}}^{n+1}(\xi_k - \xi_j)$$

and

$$U(\xi-\xi_k) = \begin{Bmatrix} (\xi-\xi_k)^0 & \text{, for } \xi > \xi_k \\ 0 & \text{, for } \xi \le \xi_k \end{Bmatrix}$$

is a unit step function.

From (1), the cubic B-spline function is given by

$$\begin{aligned} &S_3(\xi-\xi_k) \\ &\equiv B_3(\xi;\xi_{k-2},\xi_{k-1},\xi_k,\xi_{k+1},\xi_{k+2}) \\ &= [(\xi-\xi_{k-2})^3 U(\xi-\xi_{k-2}) \\ &\quad -4(\xi-\xi_{k-1})^3 U(\xi-\xi_{k-1}) \\ &\quad +6(\xi-\xi_k)^3 U(\xi-\xi_k) \\ &\quad -4(\xi-\xi_{k+1})^3 U(\xi-\xi_{k+1}) \\ &\quad +(\xi-\xi_{k+2})^3 U(\xi-\xi_{k+2})]/6\Delta^4, \end{aligned} \tag{2}$$

where $\Delta = \xi_k - \xi_{k-1}$.

An interpolation function can be expressed in the following form

$$\hat{f}(\xi) = \sum_{k=1}^{N} c_k S_k(\xi-\xi_k) \tag{3}$$

where c_k is the coefficients to be determined from the input data, $S_k(\xi-\xi_k)$ is the spline basis function, and N is the number of given data points.

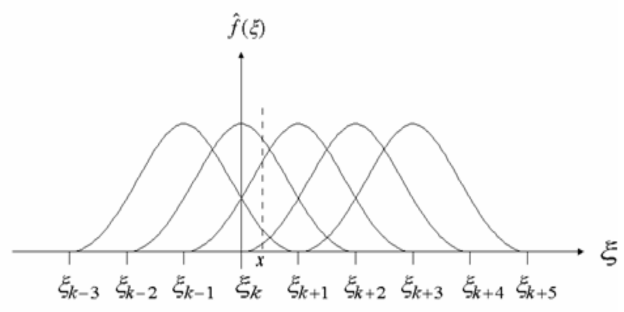


FIG. 2 $\hat{f}(\xi)$ INTERPOLATED BY CUBIC B-SPLINE

From (1)-(3) and illustrated in Fig.2., the cubic B-spline interpolation function in one-dimension (1-D) can be obtained as

$$\begin{aligned} \hat{f}(\xi) = \{&c_{k-1}[(\xi-\xi_{k-3})^3 - 4(\xi-\xi_{k-2})^3 + 6(\xi-\xi_{k-1}) \\ &-4(\xi-\xi_k)^3] + c_k[(\xi-\xi_{k-2})^3 - 4(\xi-\xi_{k-1})^3 \\ &+6(\xi-\xi_k)^3] + c_{k+1}[(\xi-\xi_{k-1})^3 - 4(\xi-\xi_k)^3] \\ &+c_{k+2}[(\xi-\xi_k)^3]\}/6\Delta^4, \end{aligned} \tag{4}$$

where c_{k-1}, c_k, c_{k+1} and c_{k+2} are the coefficients to be determined from the input data.

Let $\xi = \xi_k + x\Delta$, where $0 \le x \le 1$. The cubic B-spline interpolation function in (4) can be written as

$$\begin{aligned} &\hat{f}(\xi_k + x\Delta) \\ &= \{c_{k-1}[(3+x)^3 - 4(2+x)^3 + 6(1+x)^3 - 4x^3] \\ &\quad + c_k[(2+x)^3 - 4(1+x)^3 + 6x^3] \\ &\quad + c_{k+1}[(1+x)^3 - 4x^3] \\ &\quad + c_{k+2}x^3\}/6\Delta \\ &= \{x^3(c_{k+2} - 3c_{k+1} + 3c_k - c_{k-1}) \\ &\quad + x^2(3c_{k+1} - 6c_k + 3c_{k-1}) \\ &\quad + x(3c_{k+1} - 3c_{k-1}) \\ &\quad + (c_{k+1} + 4c_k + c_{k-1})\}/6\Delta. \end{aligned} \tag{5}$$

Then (5) can be used to find the interpolation at any point among sampled points. In particular, at the node point $\xi = \xi_k$, i.e., $x = 0$, (5) becomes

$$\hat{f}(\xi_k) = (c_{k+1} + 4c_k + c_{k-1}) / 6\Delta. \quad (6)$$

In other words, the 1-D filter of cubic B-spline function in (6) is [1, 4, 1] / 6.

Based on the definition of the two-dimensional (2-D) interpolation function [10], the cubic B-spline interpolation function can be extended from the 1-D interpolation function to the 2-D interpolation function.

Let $\hat{f}(\xi,\eta)$ be the 2-D cubic B-spline interpolation function at the point ($\xi = \xi_k + x\Delta$, $\eta = \eta_l + y\Delta$), where $0 \le x \le 1$ and $0 \le y \le 1$. By (5), one obtains

$$\begin{aligned}\hat{f}(x,y) = \{ & \hat{f}_{l-1}(x)[(3+y)^3 - 4(2+y)^3 + 6(1+y)^3 \\ & -4y^3 + \hat{f}_l(x)[(2+y)^3 - 4(1+y)^3 + 6y^3] \\ & + \hat{f}_{l+1}(x)[(1+y)^3 - 4y^3] + \hat{f}_{l+2}(x)y^3\} / 6\Delta\end{aligned} \quad (7)$$

In particular, at the node point (ξ_k, η_l), i.e., $x = 0$ and $y = 0$, (7) gives

$$\begin{aligned}\hat{f}(\xi_k,\eta_l) = \{ & (c_{k-1,l-1} + 4c_{k,l-1} + c_{k+1,l-1}) \\ & + 4(c_{k-1,l} + 4c_{k,l} + c_{k+1,l}) \\ & + (c_{k-1,l+1} + 4c_{k,l+1} + c_{k+1,l+1})\} / 36\Delta^2,\end{aligned} \quad (8)$$

for all $k = 1,2,\cdots,K$ and $l = 1,2,\cdots,L$.

In other words, the 2-D filter of cubic B-spline function in (8) is [1, 4, 1; 4, 16, 4; 1, 4, 1] / 36.

The Proposed Enhancement Scheme

In this paper, the proposed enhancement algothrim applies the philosophy of the cubic B-spline filter to improve the non-linear image enhancement method [2] which is shown in Fig.3.

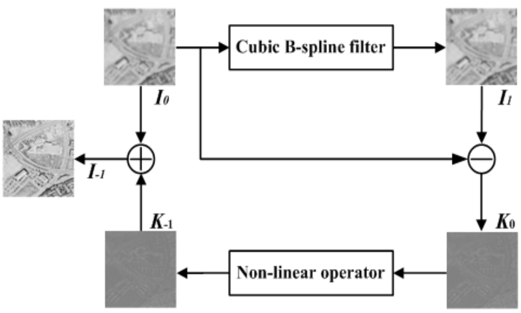


FIG. 3 THE PROPOSED NON-LINEAR IMAGE ENHANCEMENT METHOD

The low-frequency image I_1 is obtained from the input-blurred image I_0 using the cubic B-spline filter, and the high-frequency image K_0, called the residual image, is obtained by subtracting the low-frequency image I_1 from the input-blurred image I_0, i.e.,

$$K_0 = I_0 - I_1. \quad (9)$$

By [2], the enhanced image I_{-1} is generated as the sum of the input-blurred image I_0 and the predicted higher-frequency image K_{-1}; that is,

$$I_{-1} = I_0 + K_{-1}, \quad (10)$$

where $K_{-1} = NL(K_0)$ is a non-linear operator of K_0, which includes both scaling and clipping steps, defined as follows:

$$NL(K_0) = s \times Clip(K_0), \quad (11)$$

where the scaling constant s ranges in between 1 and 10 and $Clip(x)$ is given by

$$Clip(x) = \begin{cases} T, & if\ x > T \\ x, & if\ -T \le x \le T \\ -T, & if\ x < -T \end{cases} \quad (12)$$

where x is the pixel of the high-frequency image K_0,

$$T = c \times K_{0\max}, \quad (13)$$

where $K_{0\max}$ is the maximum pixel of the high-frequency image K_0 and the clipping constant c ranges in between 0 and 1.

After a non-linear operator, the higher-frequency image K_{-1} can be utilized to enhance the input-blurred image I_0.

Then, using the description in [2], the values of c and s parameters are estimated as follows. Let the standard deviation σ_0 of input image I_0 be 0.9, the standard deviation σ_{-1} of enhanced image I_{-1} be 0.45, and the low-pass (cubic B-spline) filter LF is normal distribution, whose standard deviation σ_{LF} be 1. Through the low-pass filter LF, the low-frequency image I_1 is acquired. So that the standard deviation σ_1 of I_1 is 1.345 ($\sigma_1{}^2 = \sigma_0{}^2 + \sigma_{LF}{}^2 = 1.81$). The residual image K_0 in (9) is obtained by subtracting I_1 from I_0. Through the normalized Gaussian filter, the

ErrorFunction is generated. That is,

$$K_0 = Erf(x/(\sigma_0)) - Erf(x/(\sigma_1)) \,. \tag{14}$$

Therefore, the maximum value can be obtained as

$$Erf'(x\max/(\sigma_0)) - Erf'(x\max/(\sigma_1)) = 0 \,. \tag{15}$$

or

$$I(\sigma_0)_{x\max} - I(\sigma_1)_{x\max} = 0 \,, \tag{16}$$

and

$$x\max = sqrt(2\log(\sigma_1/\sigma_0)/(1/(\sigma_0)^2 - 1/(\sigma_1)^2)) \tag{17}$$

It is followed from [2] that the solution of (14) and (17) can be confirmed. Thus, one parameter combination of c = 0.45, s = 3 from the theoretical evaluation and the other parameter combination of c = 0.4, s = 5 from the estimation analysis are proposed for the non-linear

A 3-D Down-scaling for Video Coding

In order to obtain a low bit-rate video, a new type of 3-D down-scaling scheme [11] is presented for video coding. In general, the process of decreasing the data rate is called decimation while the process of increasing data samples is called interpolation [3]. This 3-D down-scaling scheme applies a 3-D decimation with a compression ratio of 8 to 1 as the pre-processing step of the encoder. As a consequence, a 3-D interpolation with a ratio of 1 to 8 is used for the post-processing step of the decoder.

A 3-D Linear Decimated Scheme

Let t_1, t_2 and t_3 be the integer indices and n_1, n_2 and n_3 are also integers. The 3-D decimated scheme [12] takes an video $X(t_1,t_2,t_3)$ as an input and produces an output of $Y(t_1,t_2,t_3)$ by a factor of 2 in each dimension as follows:

$$Y(t_1, t_2, t_3) = \operatorname{avg}\left(\sum_{i_1=0}^{1}\sum_{i_2=0}^{1}\sum_{i_3=0}^{1} X(2t_1+i_1,\, 2t_2+i_2,\, 2t_3+i_3)\right) \quad \text{for } 0 \le t_i \le n_i - 1, \quad i = 1,2,3. \tag{18}$$

where $avg(\bullet)$ is returns the average (arithmetic mean) of a set of numeric values. Fig.4. shows the down-sampling of pre-processing stage using the 3-D linear method. In this figure, a simplified example with two adjacent 4×4 frames, that is, frame N and frame N+1, is illustrated. Firstly, two adjacent 4×4 frames are divided into four separate 2×2 groups as depicted in Fig.4(a). Then, each 2×2 group is calculated by (18) to obtain a corresponding decimated data of $Y(t_1,t_2,t_3)$ as shown in Fig.4(b).

A 3-D Linear Interpolated Scheme

It is followed from [9] that a symmetric extension scheme is used to solve the boundary condition problem to compute the 3-D interpolation at both boundaries of the image. To illustrate this, we consider the 1-D case, for example, in Fig.5, if X_k for k = 0,1,2, … ,7 is original data, then $X_{-k} = X_k$ and $X_{7+k} = X_{7-k}$ are extended from the left and right boundaries of X_k, respectively.

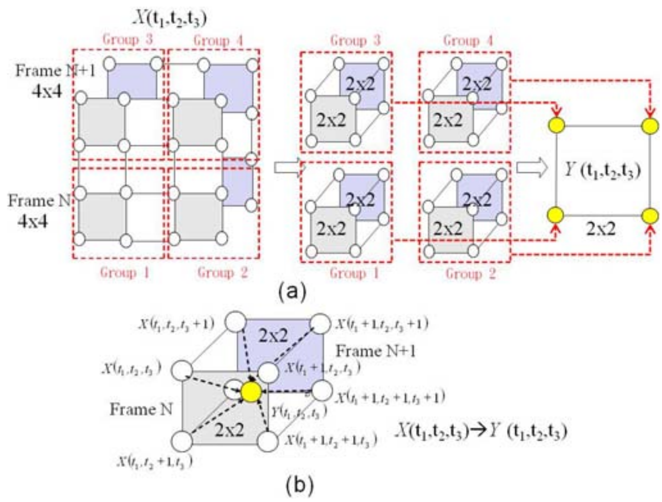


FIG. 4 A 3-D LINEAR DECIMATED SCHEME

$$\cdots X_{-4}\, X_{-3}\, X_{-2}\, X_{-1} \quad X_0\, X_1\, X_2\, X_3\, X_4\, X_5\, X_6\, X_7 \quad X_8\, X_9\, X_{10}\, X_{11} \cdots$$

FIG. 5 SYMMETRIC EXTENSION OF IMAGE DATA

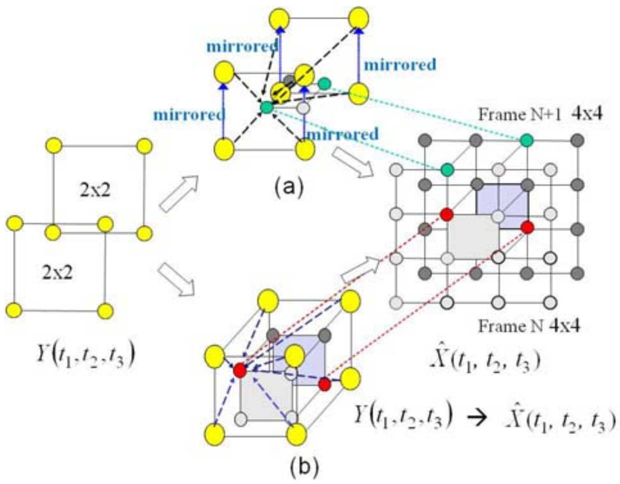


FIG. 6 A 3-D LINEAR INTERPOLATED SCHEME

Using the decimated video $Y(t_1,t_2,t_3)$ obtained from (18), the 3-D reconstructed video [12] can be calculated

by a linear interpolation shown in Fig.6. and given by

$$\hat{X}(t_1, t_2, t_3) = \sum_{k_1=0}^{1}\sum_{k_2=0}^{1}\sum_{k_3=0}^{1} Y(k_1, k_2, k_3) R(t_1 - 2k_1, t_2 - 2k_2, t_3 - 2k_3) \quad (19)$$

$$\text{for } 0 \le t_i \le 3, \quad i = 1,2,3.$$

where $R(t_1 - 2k_1, t_2 - 2k_2, t_3 - 2k_3)$ is the 3-D linear functions defined as

$$R(t_1, t_2, t_3) = R(t_1)R(t_2)R(t_3) \quad (20)$$

and $R(t)$ is the 1-D linear function given by

$$R(t) = \begin{cases} 1 - |t|/2 & , |t|/2 \\ 0 & , otherwise \end{cases} \quad (21)$$

That is, in the 3-D linear interpolated scheme, using the reconstructed values at the decimated video points $Y(t_1, t_2, t_3)$, the reconstructed video points $\hat{X}(t_1, t_2, t_3)$ between the decimated video points are obtained by the use of (19). A simplified example with the 2×2 decimated video points of $Y(t_1, t_2, t_3)$ that are interpolated is depicted in Fig.6(b) and in Fig.6(a) a mirrored method based on the above symmetric extension scheme is used to solve the boundary condition problems of computing the $\hat{X}(t_1, t_2, t_3)$ in (19). Finally, the two 4×4 reconstructed video frames are obtained as shown in Fig.6.

Computation Complexity

In order to illustrate the computation complexity of the proposed 3-D down-scaling scheme, the number of multiplication and addition/subtraction/shift is estimated in Table 1. In this table, the operations are categorized into two groups: one is multiplication operation and the other is addition, subtraction, and shift operation. Obviously, the estimated operation numbers of the proposed 3-D down-scaling scheme in both dicimation and interpolation are very compact and thus the proposed 3-D down-scaling scheme can be realized quite easily in real time.

TABLE 1. NUMBER OF OPERATIONS FOR 3-D DOWN-SCALING SCHEME.

Function	Unit	Multiplication	Addition, Subtraction, Shift	Estimation Basis
3-D Linear Decimated Scheme	(4×4)×2 blocks →2×2 block	0	36	3-D decimation of spatial (vertical, horizontal) and temporal directions are performed independently. vertical direction: (4×2)×2=16 addition horizontal direction: (2×2)×2=8 addition temporal direction: 2×2=4 addition rounding: 2×2=4 addition divided by 8: 2×2=4 3-bit shift
3-D Linear Interpolated Scheme	2×2 block→ (4×4)×2 blocks	0	120	3-D interpolation of spatial (vertical, horizontal) and temporal directions are performed independently. vertical direction: 4×2=8 addition horizontal direction: 4×4=16 addition temporal direction: (4×4)×2=32 addition rounding: (4×4)×2=32 addition divided by 8: (4×4)×2=32 3-bit shift

Video Coding Using Non-linear Image Enhancement Compensation

The 3-D down-scaling for video coding provides a better performance at a lower bit-rate transmission, however, this method causes the blurred problem of decoded image. In this section, the proposed non-linear image enhancement compensation with the cubic B-spline filter is used to improve the decoded quality of the 3-D down-scaling for video coding shown in Fig.7. In this figure, this algorithm applies the 3-D linear decimation (3D-LI↓) as the encoder, and the 3-D linear interpolation (3D-LI↑) as the decoder for video coding. As a consequence, the proposed non-linear image enhancement compensation with cubic B-spline filter is used for the post-processing step of this decoder.

For this algorithm, an original video (image) sequence in the RGB color space is converted into another preliminary sequence in YUV [4, 5] color space prior to the 3D-LI↓ processing. The size of the original RGB video sequence in 352×288 resolution is assumed to be 352×288×3×30 bytes for a period N = 30 frames. After color-space conversion, one set of 352×288×30 bytes is used for Y, and two sets of 176×144×30 bytes are used for the U and V video sequences. In the proposed 3D-LI↓ procedure, the input video sequence is a Y video sequence of size 352×288×30 bytes, and the output

video sequence is a decimated video sequence of size 176×144×15 bytes. For the U and V video sequences, the input video sequence has 176×144×30 bytes so that the output video sequence to be decimated is 88×72×15 bytes. At the end of the encoder, the three separate Y, U, and V decimated video sequences are combined into one YUV decimated video sequence.

In addition, in the decoder, there are two processes used that are reversed in the encoding steps. In the first step, the YUV received (decimated) video sequence is separated into three separate Y, U, and V decimated video sequences. Then, the proposed 3D-LI↑ process uses the linear interpolation to reconstruct the video sequences. After this interpolation, the size of the Y video sequence is therefore converted from 176×144×15 bytes to 352×288×30 bytes, and the U and V video sequences are increased from 88×72×15 bytes to 176×144×30 bytes. In the second step, the proposed image enhancement compensation method, described in Section 3, is used for the Y video sequence only, and it is not used for the U and V video sequences. Furthermore, the three Y, U, and V video sequences are combined again into one YUV format. Finally, it is followed from [4, 5] that this YUV video sequence is converted into the reconstructed RGB video sequence with size 352×288×3×30 bytes.

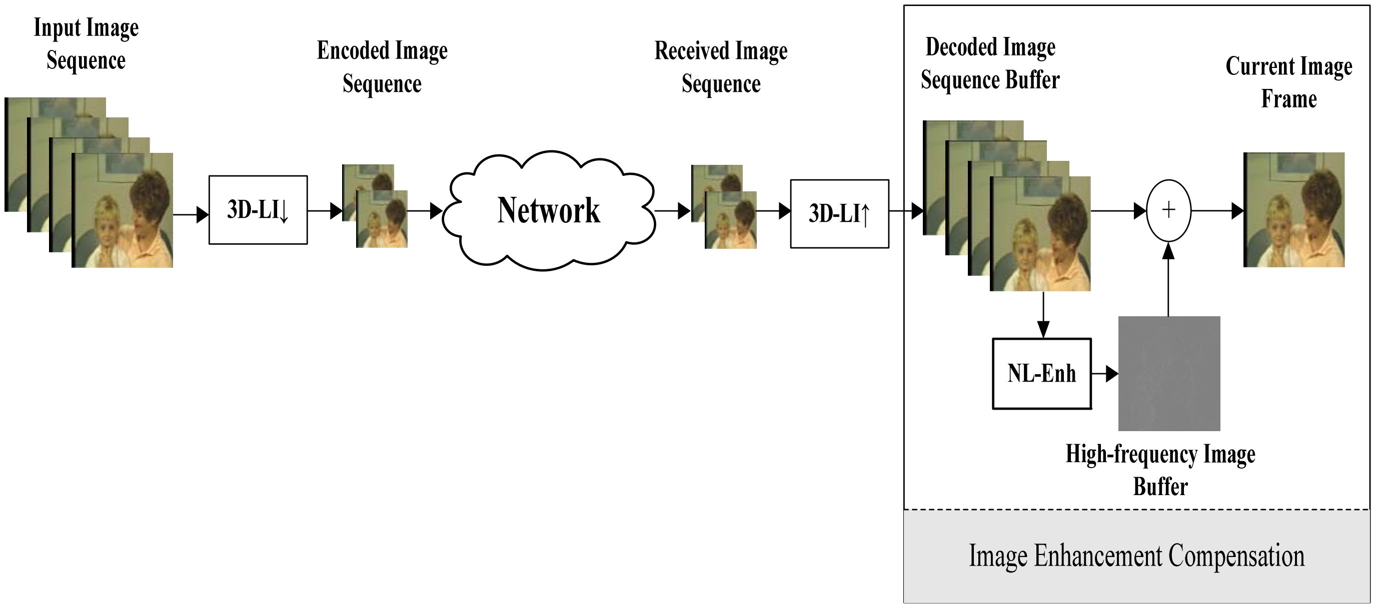


FIG. 7 VIDEO CODING USING NON-LINEAR IMAGE ENHANCEMENT COMPENSATION

Computer Simulation

Let $X(i,j)$ and $\hat{X}(i,j)$ be the original (input) and reconstructed images, respectively, and let i, j for $0 \le i \le M-1$ and $0 \le j \le N-1$ be the index numbers that determine the vertical and horizontal positions of pixels in the images. The mean square error (MSE) between $X(i,j)$ and $\hat{X}(i,j)$ is given by

$$MSE = \left(\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| X(i,j) - \hat{X}(i,j) \right|^2 \right) / (M \times N) . \qquad (22)$$

Thus the peak signal-to-noise ratio (PSNR) between $X(i,j)$ and $\hat{X}(i,j)$ is defined by

$$PSNR(dB) = 10 \log_{10} \left(b^2 / MSE \right) , \qquad (23)$$

where b is the largest value of the image signal (typically 255 for 8-bits of gray level).

In this section, some experimental results of three gray images (Aerial, Baboon, Barbara) in 512×512 resolution, three color images (Lena, Peppers, Sailboat) in 512×512 resolution and three video sequences (Mother, Stefan, Table) in 352×288 resolution are presented by computer simulations. First of all, these gray and color images are blurred to low-resolution images, repectively, then the proposed method and the methods in [1] and [2] are used to enhance these low-resolution images, in addition, the PSNR of these experimental results are compared in Table 2 and Table 3. Furthermore, the 3-D down-scaling for video coding is used along with the proposed method and the methods in [1] and [2] to enhance the reconstructed images of the above three video sequences, finally, the PSNR of these experimental

results are also compared in Table 4. Obviously, in Tables 2, 3 and 4, the quality of reconstructed images using the proposed method is better than the methods in [1] and [2].

Note that in clipping and scaling parameters, s=5, c=0.4 are selected for both gray and color images, but for 3-D down-scaling video coding s=3, c=0.45 are chosed, because these parameters get better PSNR results.

TABLE 2. PSNR (DB) OF GRAY ENHANCED IMAGE OF SIZE 512×512 FOR GAUSSIAN[1], FSD[2], AND PROPOSED METHODS.

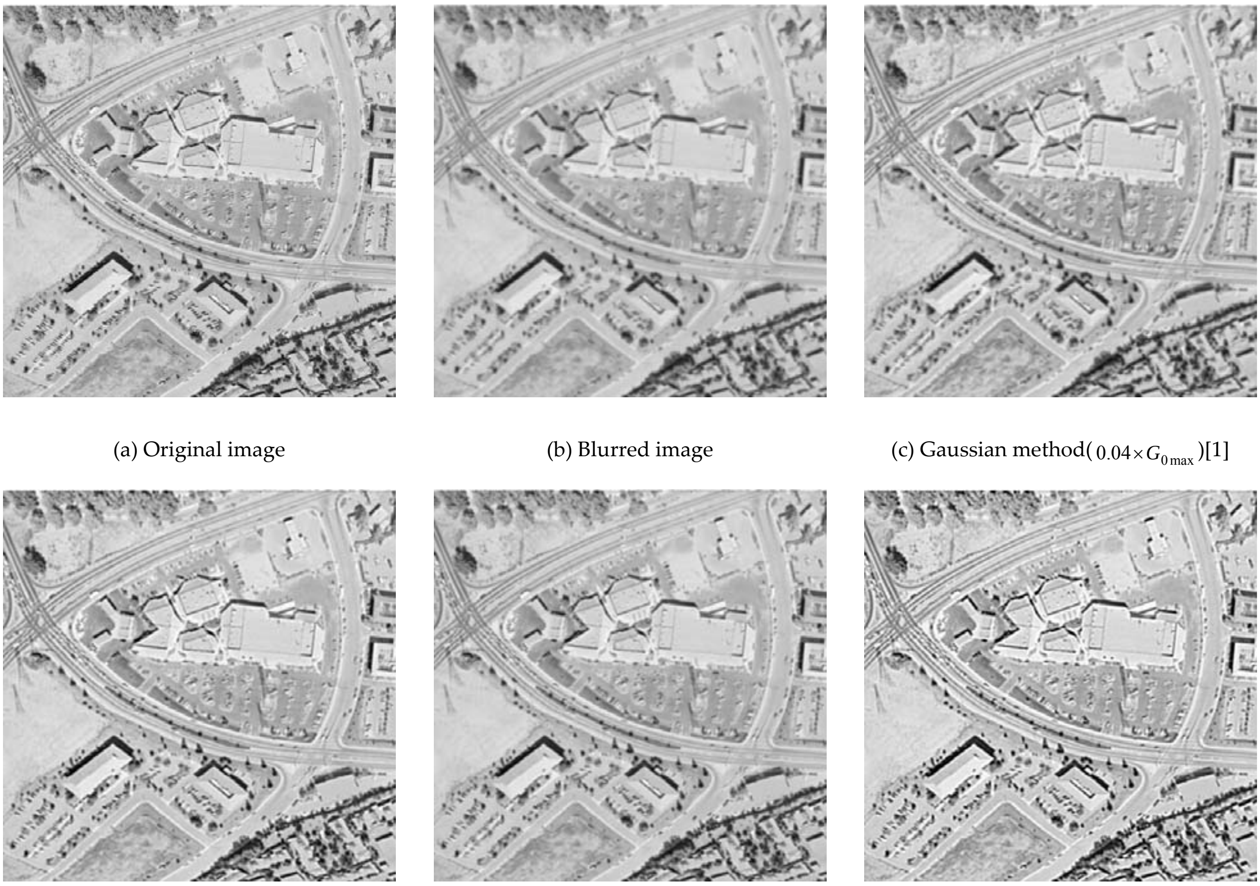
Image Name	Blurred Image	Gaussian method[1]	FSD method [2]		Proposed method
		$0.04 \times G_{0\max}$	s=3,c=0.45	s=5,c=0.4	s=5,c=0.4
Aerial	25.57	25.81	27.94	25.12	28.46
Baboon	21.88	22.20	23.49	22.59	23.50
Barbara	24.68	24.54	25.58	24.87	25.72

TABLE 3. PSNR (DB) OF COLOR ENHANCED IMAGE (Y) OF SIZE 512×512 FOR GAUSSIAN[1], FSD[2], AND PROPOSED METHODS.

Image Name	Blurred Image	Gaussian method[1]	FSD method [2]		Proposed method
		$0.04 \times G_{0\max}$	s=3,c=0.45	s=5,c=0.4	s=5,c=0.4
Lena	31.97	30.79	34.36	30.79	34.81
Peppers	29.80	28.97	31.18	29.28	31.44
Sailboat	27.62	27.43	29.66	26.97	29.93

TABLE 4. PSNR (DB) OF ENHANCED VIDEO SEQUENCE (Y) OF SIZE 352x 288 BY 3-D DOWN-SCALING FOR GAUSSIAN[1], FSD[2], AND PROPOSED METHODS.

Sequence Name	3D-LI ↓ / 3D-LI ↑	Gaussian method[1]	FSD method [2]		Proposed method
		$0.04 \times G_{0\max}$	s=3,c=0.45	s=5,c=0.4	s=3,c=0.45
Mother	33.97	32.74	33.74	29.93	35.19
Stefan	21.43	21.60	21.93	20.64	22.04
Table	24.07	24.26	24.54	22.92	24.94



(a) Original image (b) Blurred image (c) Gaussian method($0.04 \times G_{0\max}$)[1]

(d) FSD method(s=3,c=0.45)[2] (e) FSD method(s=5,c=0.4)[2] (f) Proposed method(s=5,c=0.4)

FIG.8 COMPARISON OF SUBJECTIVE QUALITY OF ENHANCED GRAY AERIAL IMAGE.

Finally, in Figs.8, 9 and 10, clearly, the enhanced image by means of the proposed method indicates a better subjective quality than that by other enhanced methods. Though the FSD method is the clearest algorithm, it causes the ring effect for the enhanced image because the higher-frequency component is enhanced too much, and the Gaussian method is inferior in performance to the proposed method and

the FSD method.

(a) Original image (b) Blurred image (c) Gaussian method($0.04 \times G_{0\max}$)[1]

(d) FSD method(s=3,c=0.45)[2] (e) FSD method(s=5,c=0.4)[2] (f) Proposed method(s=5,c=0.4)

FIG.9 COMPARISON OF SUBJECTIVE QUALITY OF ENHANCED COLOR SAILBOAT IMAGE(ZOOM IN).

(a) Original image (b) Blurred image (c) Gaussian method($0.04 \times G_{0\max}$)[1]

(d) FSD method(s=3,c=0.45)[2] (e) FSD method(s=5,c=0.4)[2] (f) Proposed method(s=3,c=0.45)

FIG.10 COMPARISON OF SUBJECTIVE QUALITY OF ENHANCED COLOR MOTHER IMAGE (VIDEO SEQUENCE).

Conclusions

In this paper, the proposed algorithm uses the cubic B-spline filter and the non-linear image enhancement compensation to improve the blurred image and video sequence. Such a new method is not only efficient but also superior in performance to the other image enhancement methods. Experimental results also show that the proposed method yields a better subjective quality and objective PSNR than other non-linear image enhancement methods for the reconstructed image. Furthermore, the proposed algorithm can be realized quite easily in real time and also can be used with the scalable video coding [4] for the digital broadcast TV/HDTV, IPTV, and video-phone applications.

ACKNOWLEDGMENT

This work was supported by the National Science Council, R.O.C., under Grant NSC 96-2221-E-251-004. The author would like to thank Cheng-Ying Wu, Kun-Rong Shieh, Ying-Lun Tang and Ya-Chun Huang for their valuable contributions.

REFERENCES

Greenspan, H., "Multi-resolution image processing and leaming for texture recognition and image enhancement," PhD thesis, California Inst. of Technol., 1994.

Greenspan, H., Anderson, C. H., and Akber, S., "Image enhancement by nonlinear extrapolation in frequency space," IEEE Trans. on Image Processing, vol. 9, no. 6, pp. 1035-1048, June 2000.

Hou, H. S., and Andrews, H. C., "Cubic spline for image interpolation and digital filtering," IEEE Trans. Acoustics Speech Signal Processing, vol. 26, pp. 508–517, Dec. 1978.

Li, Z. N., and Drew, M. S., Fundamentals of Multimedia, Prentice-Hall, 2004.

Pratt, W. K., Digital Image Processing, second edition, John Wiley & Sons, Inc., New York, 1991.

Unser, M., Aldroubi, A., and Eden, M., "B-spline signal processing: Part I - Theory," IEEE Transactions on Signal Processing, vol. 41, no. 2, pp. 821-833, Feb. 1993.

Unser, M., Aldroubi, A., and Eden, M., "B-spline signal processing: Part II - Efficient design and applications," IEEE Transactions on Signal Processing, vol. 41, no. 2, pp. 834-848, Feb. 1993.

Unser, M., "Splines – A perfect fit for signal and image processing," IEEE Signal Processing Magazine, vol. 16, no. 6, pp. 22-38, Nov. 1999.

Wang, L. J., Hsieh, W. S., and Truong, T. K., "A fast computation of 2D cubic-spline interpolation," IEEE Signal Processing Letters, vol.11, no.9, pp.768-771, Sept. 2004.

Wang, L. J., Shieh, K. R., and Tang, Y. L., "A non-linear image enhancement method by spline basis filter," in Proc. of International Computer Symposium 2006 (ICS 2006), Taipei, Taiwan, Dec. 4-6, 2006.

Wang, L. J., and Huang, Y. C., "An improved non-linear image enhancement method for video coding," in Proc. of the Second International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008), pp. 79-84, Barcelona, Spain, March 4-7, 2008.

Wang, L. J., "Video coding with 3-D dynamic resolution conversion and rate-distortion," Communications in Information Science and Management Engineering, vol. 2, no. 9, pp. 18-25, Sept. 2012.

Yang, Y., and Li, B., "Non-linear image enhancement for digital TV applications using Gabor filters," in Proc. of IEEE International Conference on Multimedia and Expo, July 6-8, 2005.

Yuan, S., Taguchit, A., and Kawamata, M., "Arbitrary scale image enlargement with the prediction of high frequency components," in Proc. of IEEE International Symposium on Circuits and Systems, vol. 6, pp.6264-6267, May 23-26, 2005.

Lung-Jen Wang received the Ph.D. degree in Computer Science and Engineering from the National Sun Yat-Sen University, Taiwan, 2001. He was the Chairman of the Department of Information Technology, National Pingtung Institute of Commerce, Pingtung, Taiwan, from 2005 to 2008, where he was the Secretary General from 2008 to 2010 and the Acting President from 2010 to 2011.

Currently, he is an Associate Professor in the Department of Computer Science and Information Engineering at National Pingtung Institute of Commerce, Taiwan. His primary research interests are in the areas of image processing, multimedia network and real-time system.